Stratification in P2P networks, Application to BitTorrent

Anh-Tuan Gai, Fabien Mathieu, Fabien de Montgolfier, Julien Reynier

Stratification in P2P Networks: Application to BitTorrent

Anh-Tuan Gai
INRIA
domaine de Voluceaux
78153 Le Chesnay cedex, France
anh-tuan.gai@inria.fr

Fabien Mathieu
Orange Labs
38–40, rue du général Leclerc
92130 Issy les Moulineaux, France
fabien.mathieu@orange-ftgroup.com

Fabien de Montgolfier
LIAFA–Université Paris 7
175, rue du Chevaleret
75013 Paris France
fm@liafa.jussieu.fr

Julien Reynier
Motorola Labs
Parc les Algorithmes, Saint Aubin
91193, Gif sur Yvette Cedex, France
julien.reynier@normalesup.org

Abstract

We consider a model for decentralized collaborative networks that is based on stable matching theory. This model is applied to systems with a global ranking utility function, which admits a unique stable configuration. We study the speed of convergence and analyze the stratification properties of the stable configuration, both when all collaborations are possible and for random possible collaborations.

As a practical example, we consider the BitTorrent Tit-for-Tat policy. For this system, our model provides an interesting insight into peer download rates and a good understanding of upload settings strategy.

1 Introduction

Motivation Collaboration is a new paradigm for managing QoS in large scale systems. A system is said to be collaborative when participating peers offer resources to each other so that they reach their goal. Apart from well-known content distribution applications [3, 4], collaborative systems appear in numerous applications such as distributed computing, online gaming, or decentralized backup. The common property of such systems is that participating peers exchange resources. The underlying mechanism provided by protocols for such applications consists in selecting which peers to collaborate with in order to maximize one's peer benefit with regards to its personal interest. This mechanism generally uses a utility function taking local information as input. One can ask if this approach can provide desirable properties for the resulting collaboration graph, like scalability and reliability.

For instance, the well-known protocol BitTorrent [3] implements a Tit-for-Tat (TFT) exchange policy. More precisely, each node *knows* a subset of all other nodes of the system and collaborates with the best ones from its point of view: it uploads to the contacts it has most downloaded from in the last 10 seconds. In other words, the utility of peer i for node j is equal to the quantity of data peer j has downloaded from i (in last 10"). The main interest in using the TFT policy is incentive to cooperate. The nature of the utility function then leads to a clustering process which gather peers with similar upload performances together, called *stratification*.

Recently, much research has been devoted to the study of this phenomenon. So far, however, while it has been measured and observed by simulations, it has not been formally proved. Understanding stratification is a first step towards a better comprehension of the impact of the utility function on system behavior. A theoretical framework to analyze and compare different utility functions is needed, since choosing a utility function that best suits a given application is quite difficult. And it is not clear whether the utility functions implemented lead to desirable properties.

Contribution In a previous work [5], we introduced a generic framework that allows an instantiation of (known and novel) utility functions that model collaboration. In the present paper, we first apply this framework to model systems with a global ranking utility function, where each peer has an intrinsic value. BitTorrent TFT policy is a canonical example of such systems: the ranking function is the bandwidth offered. These systems possess a unique stable configuration. We simulate the speed of convergence with and without *churn* (arrivals ans departures). The good convergence properties observed validate the interest of studying

the stable configuration properties.

Second, we consider a toy model of fully connected networks where every peer can collaborate with every other peers, and we study the stratification in the stable configuration. If every peer tries to collaborate with the same number of peers, we observe disjoint clustering. But with a variable number of collaborations per peer, clustering turns into strong stratification.

Third, we describe stratification in random graphs. For Erdös-Rényi graphs, the distribution of collaborating peers has a fluid limit. This limiting distribution shows that stratification is a scalable result.

Lastly, we propose a practical application of our results to the BitTorrent TFT policy. With the assumption that content availability is not a bottleneck in a BitTorrent swarm, our model leads to an interesting characterization of the download rate a peer can expect as a function of its upload rate. This description leads to a better understanding of download/upload correlations, and is a first step to analyze possible strategies to optimize the download for a given upload rate.

Roadmap In Section 2 we define our model and notations. Section 3 gives the related work, with an emphasis on the stable matching framework on which this paper is based. Section 4 presents a study on convergence issues. We describe stratification in a complete neighborhood graph in Section 5 and in random graphs in Section 6. Section 7 discusses the application of our results to BitTorrent and Section 8 concludes the paper.

2 Model and notations

Our model is based on a dynamical version of the stable b-matching problem [2] where each peer i has a *global mark* $S(i)$. More precisely we consider networks where there exists a total order on peers (based on bandwidth capacities for instance).

Each peer i ranks a subset of participating peers (or all peers) with respect to (w.r.t.) global marks. We denote by preference list of peer i, the resulting ordered subset. If peer i does not belong to peer j's preference list, i is said to be unacceptable for j. We thus introduce an *acceptance graph* to represent compatibilities. An edge $\{i, j\}$ belongs to the acceptance graph if, and only if (iff) i belongs to j's preference list and j belongs to i's preference list.

In our model, as in the b-matching problem, each peer tries to improve its own payoff by collaborating (being matched) with its best neighbors w.r.t. marks and the acceptance graph. We denote by *configuration* or *matching* the subgraph of the acceptance graph that represents the effective collaboration between peers at a given instant. The degree of a peer i in a configuration is bounded by $b(i)$, its collaboration quota.

A *blocking pair* for a given configuration is a set of two peers unmatched together wishing to be matched together (even if it means dropping one of their current collaborations). A configuration without blocking pair is said to be *stable*. In a stable configuration, a single peer cannot improve its situation: it is a Nash equilibrium.

We introduce the concept of *initiative* to model the process by which a peer may change its mates. Given a configuration C, we say that peer i *takes the initiative* when it proposes to other peers to be its new mate. Basically, i may propose partnership to any acceptable peer. Nevertheless, only blocking pairs of C represent interesting new partnerships. If i can find such a *blocking mate*, the initiative is called *active* because it succeeds in modifying the configuration (both peers will change their set of mates). If i has already $b(i)$ established collaborations in C, it drops the worst (w.r.t. marks) to establish a new one.

To find a blocking mate, i contacts peers from its acceptance list and selects the best available (if any). We can now complete our model with initiatives: starting from any initial configuration, an instance of our model evolves because of initiatives taken by peers. For more information on initiative strategies, the reader may refer to [5].

3 Related work

Stable matching Tan [11] has shown that existence and uniqueness of stable solutions were related to preference cycles induced by the utility function used to compute preference lists. A preference cycle of length k is a set $i_1, \ldots, i_k$ of k distinct peers such that each peer of the cycle prefers its successor to its predecessor. As proved by Tan, a stable configuration exists if there is no odd preference cycle of length greater than 1. He also proved that if no even cycle of length greater than 2 exists, then the stable configuration is unique. If peers have an intrinsic value, no strict preferences cycle can occur , so a global-ranking matching problem admits one unique stable solution.

Models of BitTorrent-like systems In [8], a model of BitTorrent is derived, where the authors survey the evolution of seed and leecher numbers, and prove the existence of an equilibrium state where the actual upload rate equals the maximal upload rate. They model average peer behavior and are not concerned with stratification or share ratio issues. It is shown in [7], using a simple model of two possible bandwidths, that data replication is more efficient with heterogeneous link capacities.

4 Convergence study

Previous work The stable solution of a global ranking stable b-matching problem can be easily computed knowing the global ranking S, b and the acceptance graph. The

process is given by Algorithm 1: each peer i starts with $b(i)$ available connections. First, the best peer i_1 picks the best $b(i_1)$ peers from its acceptance list. As i_1 is the best, the chosen peers gladly accept (recall the acceptance graph is symmetric) and the resulting collaborations are stable (no blocking pair can unmatch them). Note that if there is not enough acceptable peers, i_1 may not satisfy all its connections. Peers chosen by i_1 have one less connection available. Then second best peer i_2 does the same, and so on... By immediate recurrence, all connections made are stable. When the process reaches the last peer, the connections are the stable configuration for the problem. As it was said before, all connections are not necessarily satisfied. For instance, if the last peer still has available connections when its turn comes, his connections will not be fetched, as all peers above him have by construction spent all their connections. This is, of course, a centralized algorithm, but we shall see below that decentralized algorithms work as well.

Algorithm 1: Stable configuration in global ranking

Data: Acceptance graph G with n peers, global ranking $S(i)$, maximal number of connections $b(i)$
Result: The unique stable configuration of the b-matching problem

```
Let a be a vector initialized with b
for each peer i sorted in increasing S(i) (best first) do
    for each peer j such that S(j) > S(i), sorted in increasing S(j) do
        if (i, j) ∈ G and a(i) > 0 and a(j) > 0 then
            connect (i, j)
            a(i) = a(i) − 1
            a(j) = a(j) − 1
```

We have shown in [5], that starting from any initial configuration, an instance of our model evolves towards the unique stable configuration. More precisely, in [5], we proved that in static conditions (no join or departure, constant utility function), the system eventually converges towards the stable state. But to prove this stable state is worth studying, we have to show convergence is fast in practice (Algorithm 1 is optimal in number of initiatives but difficult to implement in a large scale system) and can sustain a certain amount of churn. As a complete formal proof of this is beyond the scope of this paper, we use simulations.

Experimental setup and definitions In our simulations, peers were labeled from 1 to n (the number of peers). We choose the canonical ranking $S(i) = i$, 1 being the best peer and n the worst (if $i < j$, peer i is better than peer j). We use Erdös-Renyi loopless symmetric graphs $\mathcal{G}(n, p)$ as acceptance graphs, where p is the probability that a given edge exists (the expected degree is $d = p(n-1)$). Only 1-matching was considered.

For measuring the difference between two configurations C_1 and C_2 we use the distance

$$\delta(C_1, C_2) = \Sigma_{i=1}^{n} \|C_1(i) - C_2(i)\| . \frac{1}{n(n+1)},$$

where $C(i)$ denotes the mate of peer i in configuration C (by convention, $C(i) = n+1$ if i is unmated in C).

δ is normalized: the distance between a complete matching and the empty configuration (denoted $C_\emptyset$) is equal to 1. The *disorder* denotes the distance between the current configuration and the stable configuration.

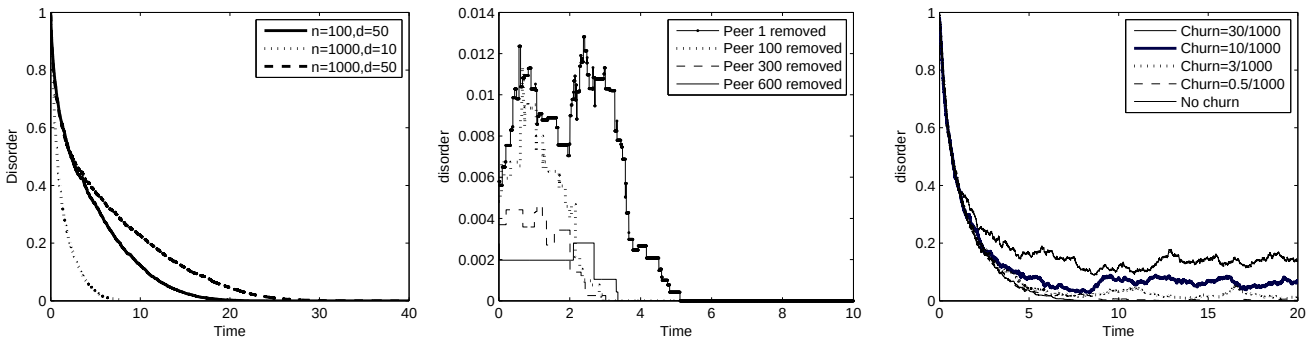
At each step of the process we simulate, a peer is chosen at random and performs an initiative (the initiative can be active or not). To compare simulations with different number n of peers, we call *time unit* a sequence of n successive initiatives (in a time unit, the expected number of initiatives per peer is one).

Simulations results A first set of simulations is made to prove a rapid convergence when the acceptance graph is static. In all simulations, the disorder quickly decreases, and the stable configuration is reached in less than nd initiatives (that is d base unit). Figure 1(a) shows convergence starting from $C_\emptyset$ for three typical parameters: $(n, d) = (100, 50)$, $(n, d) = (1000, 10)$, $(n, d) = (1000, 50)$.

Then we investigate the impact of an atomic alteration of the system. Starting from the stable configuration, we remove a peer from the system and observe the convergence towards the new stable configuration. Our simulations show big variances in convergence patterns. However, convergence always takes less than d time units and disorder is always small. Due to a domino effect, removing a good peer generally induces more disorder than removing a bad peer. Figure 1(b) shows four representative trajectories for $(n, d) = (1000, 10)$.

Finally, we investigate continuous churn. A peer can be removed or introduced in the system anytime, according to a *churn rate* parameter. Simulations show that as the churn rate increases, the system becomes unable to reach the instant stable configuration. However, the disorder is kept under control. That means the current configuration is never far from the instant stable configuration. The average disorder is roughly proportional to the churn rate (Figure 1(c) indicates typical patterns for $(n, d) = (1000, 10)$, starting from $C_\emptyset$).

All these simulations lead to the same idea: the stable configuration acts like a strong attractor in the space of possible configurations when collaborations are established using intrinsic values for judging peers. Studying the properties of stable configurations is the next step.



(a) Convergence with static acceptance graph (b) Convergence after a peer has been removed (c) Pseudo-convergence in presence of churn

Figure 1. Evolution of disorder for three basic scenarios

5 Stratification on complete acceptance graph

We first study the stable configuration in the special case where everybody is acceptable for everybody. Hence the acceptance graph is complete. This is a valid, but not scalable, assumption for small systems. Complete acceptance graph is a toy model for highlighting the stratification effect.

5.1 Clustering in constant b_0-matching

b_0-matching is an instance of the b-matching problem where everyone tries to connect to at most b_0 peers (b_0 is a constant). Since the acceptance graph is complete, the stable configuration is very simple. It consists in a sequence of complete subgraphs with $b_0 + 1$ elements starting from the best peer (the remainder, if any, is a truncated complete subgraph). For example, Figure 2(a) shows this *clustering* for the 2-matching problem on a complete graph.

As it has already been pointed out [1], full clustering in file sharing networks induces poor performances. Many designers try to produce overlay graphs with small world properties: almost fully connected, high clustering coefficient, low mean distance, and navigable such that shortest paths may be greedily found. But in file sharing networks, having a compliant overlay with nice properties (connectivity, distances, resilience) is useless if the effective collaborations graph has none of the desired properties. In our example, although the knowledge graph is a complete graph, collaboration established through global ranking scatters the graph in clusters. Hence content is sealed inside clusters, and singularities are bound to occur.

5.2 Stratification in variable b-matching

b_0-matching is not the most common case in practice. The clustering from Figure 2(a) may be a consequence of the specific parameters used. Indeed, adding one extra connection suffices to turn a set of complete subgraphs of size $b_0 + 1$ into one unique connected component (see Figure 2(b) – settings are same than for Figure 2(a) except that an extra connexion has been granted to peer 1).

In fact, both Figures 2(a) and 2(b) are not typical. If we use a random quota distribution, we generally observe many large connected components. More precisely, if we consider that b is distributed according to a rounded normal distribution $\mathcal{N}(\bar{b}, \sigma^2)$ (mean $\bar{b}$, variance σ, all samples are rounded to the nearest positive integer), we observe a surprising phase transition. As soon σ is big enough to produce heterogeneous samples ($\sigma \approx 0.15$), the average connected component size explodes, then stays almost constant. The cluster typical size after the transition seems to grow factorially with $\bar{b}$ (computed values appear in Table 1). Figure 3 shows what happens for $\bar{b} = 6$.

Factorial cluster size growth grants the existence of a giant connected component when $\bar{b}$ is large enough and n remains bounded. This solves the clustering issue.

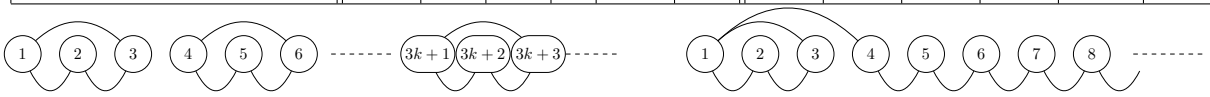
Nevertheless, distances in the obtained collaboration graph are another question. A good estimate is given by Mean Max Offset (MMO) which described the mean ranking offset between one peer and its further neighbor in the collaboration graph. The larger the MMO, the fewer hops needed to link two peers with very different intrinsic value in the same connected component. For instance, in Figure 2(a), the MMO is $\frac{5}{3}$ since the Max Offset is 1 for peers numbered $3k + 2$, and 2 for peers numbered $3k$ or $3k + 1$. Remark that in b_0-matching, MMO is easy to compute (it is enough to compute it on the $b_0 + 1$ complete graph). It converges to:

$$MMO(b_0) = \frac{1}{b_0+1}(b_0 + \ldots + \left\lceil \frac{b_0}{2} \right\rceil + \ldots + b_0) \xrightarrow[b_0 \to +\infty]{} 3/4 b_0$$

When b is variable, MMO becomes less obvious to compute. However, simulations show that MMO has the same

Table 1. Clustering and stratification properties in a complete knowledge graph.

	constant b_0-matching						normal $\mathcal{N}(\bar{b}, \sigma^2)$-matching with $\sigma = 0.2$					
b_0 or $\bar{b}$	2	3	4	5	6	7	2	3	4	5	6	7
Average Cluster Size	3	4	5	6	7	8	6	20	78	350	1800	11000
Max Mean Offset (MMO)	1.67	2.5	3.2	4	4.71	5.5	1.33	2.10	2.52	3.21	3.65	4.31



(a) $b_0 = 2$. (b) $b_0 = 2$ plus one extra connection for peer 1.

Figure 2. Stable collaboration graph for total knowledge (constant and altered b).

phase transition as the cluster size. But as cluster size explodes, MMO decreases, has shown by Figure 3 (for $\bar{b} = 6$) and Table 1.

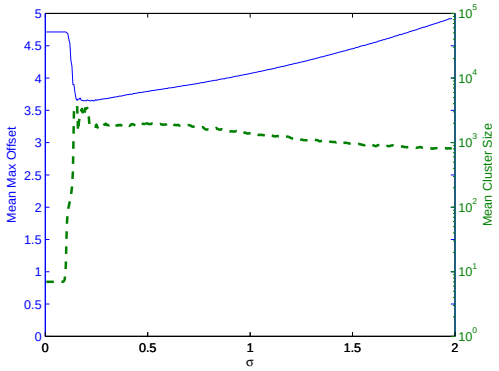


Figure 3. Mean cluster size (dotted line) and MMO (plain line) as a function of σ.

The conclusion of this first approach on complete graphs is that whereas the clustering problem can be handled, peers only collaborate with peers very close to them, which can make content diffusion ineffective. This is stratification.

6 Stratification on random acceptance graphs

We now consider the properties of the unique stable configuration in Erdös-Rényi acceptance graphs $\mathcal{G}(n,p)$ (cf Section 4). We start by working on a 1-matching model. This allows to state an *independence* assumption and related mathematical results. We then extend the equations to the b_0-matching case.

6.1 Independent 1-matching model

Let C be the unique locally optimal stable configuration. $C(i)$ denotes the mate of peer i in C. $D_{n,p}(i,j)$ is the probability that peer i is matched with peer j over all possible $\mathcal{G}(n,p)$ graphs. If there is no ambiguity, p will be omitted. Regarding n, as shown by Lemma 1, it can also be omitted, leading to the notation $D(i,j)$.

Lemma 1 *For all integers i,j,n such that $1 \leq i,j \leq n$ and for all $p \in [0,1]$,$D_{n,p}(i,j) = D_{\max(i,j),p}(i,j)$.*

Intuitively, the probability that i and j are mated does not depend on the existence of peers of worst value.

Proof: Let G be an Erdös-Rényi $\mathcal{G}(n,p)$ graph. Wlog, suppose $i < j$ ($i = j$ is trivial because $D_{n,p}(i,i) = 0$). According to Algorithm 1 (with S equal to the identity function), the decision to mate i and j only depends on the subgraph G_j of G induced by the first j peers. As G is an Erdös-Rényi graph, edges are independent, and G_j is a $\mathcal{G}(j,p)$ graph. That concludes the proof. □

An exact formula

$D(i,.)$ is the distribution of $C(i)$ (as peers may be unmated, it is not necessary a probability). Obviously, $D(i,j) = D(j,i)$ and $D(i,i) = 0$. One can observe that i is mated with j iff: $\{i,j\}$ is an edge of the acceptance graph, i is not mated with a peer better than j and j is not mated with a peer better than i. This leads to the following exact formula:

$$\begin{aligned} D(i,j) &= p\mathbb{P}\left(C(i) \not< j\right) \times \mathbb{P}\left(C(j) \not< i | C(i) \not< j\right) \\ &= p\left(1 - \sum_{k=1}^{j-1} D(i,k)\right) \mathbb{P}\left(C(j) \not< i | C(i) \not< j\right) \end{aligned} \quad (1)$$

Using equation (1), we now can prove that for $p > 0$, $D(i,.)$ is asymptotically a probability.

Lemma 2 $\forall p \in]0,1], \forall i \in \mathbb{N}^*, \sum_{k=1}^{\infty} D(i,k) = 1$.

Intuitively, each peer eventually finds a stable mate when the network grows.

Proof: We first show that $\mathbb{P}\left(C(j) \not< i | C(i) \not< j\right)$ does not go to 0 as j increases. For all $j > i$, conditioning on $E_i := \{C(1), ..., C(i-1)\}$,

$$\mathbb{P}(C(j) \not< i | C(i) \not< j | E_i) = \begin{cases} \text{empty conditioning if } i \in E_i \\ 0 \text{ if } j \in E_i \text{(and } i \notin E_i\text{)} \\ x \geq p \text{ if } j \notin E_i \text{ and } i \notin E_i. \end{cases}$$

The last inequality holds because if $j \notin E_i$ and $i \notin E_i$, then knowing that $C(i) \not< j$, i and j are linked if and only if

there exists an edge between both. Since $C(j) = i$ implies $C(j) \not< i$, the inequality is satisfied.

Next, we show that $\mathbb{P}(j \in E_i | i \notin E_i)$ does not tend to 1 when j tends to infinity: for some $k < i$, the function $j \to \mathbb{P}(C(k) = j | i \notin E_i)$ gives probabilities of disjoint events so that $\sum_{j=1}^{\infty} \mathbb{P}(j \in E_i | i \notin E_i) \leq i - 1$; the general term thus tends to 0 and certainly not to 1.

We can now prove that $D(i, .)$ is a probability: for any given i, $D(i, j)$ are the probabilities of disjoint events. Thus $D(i, j) \xrightarrow[j \to \infty]{} 0$. From formula (1) we deduce $\sum_{k=1}^{j-1} D(i, k) \xrightarrow[j \to \infty]{} 1$. □

Approximation: independent 1-matching model Hereinafter we shall adopt the following assumption:

Assumption 1 *These two events are independent:*

- *peer i is not with a peer better than j,*
- *peer j is not with a peer better than i,*

Assumption 1 is reasonable when p is small (so the probability that i and j have a common neighbor is very low). Then (1) can be approximated by:

$$D(i, j) = p \left(1 - \sum_{k=1}^{j-1} D(i, k)\right) \left(1 - \sum_{k=1}^{i-1} D(j, k)\right) \quad (2)$$

This formula can easily be computed by dynamic programming, as shown in Algorithm 2 (see Algorithm 3 for the b_0-matching case).

Algorithm 2: Independent 1-matching probability

```
Data: Erdös-Rényi parameters n, p
Result: D(i,j) the probability user i chooses user j
D ← zeros(n, n)
for i = 1 to n do
    for j = i + 1 to n do
        Compute D(i,j) using (2)
        D(j,i) ← D(i,j)
```

Formula (2) is not exact, as we can verify for the best three peers: exact computation gives $D_{\text{exact}}(1, 2) = p$ (probability that $\{1, 2\}$ is acceptable), $D_{\text{exact}}(1, 3) = p(1 - p)$ ($\{1, 3\}$ is acceptable, $\{1, 2\}$ is not), and $D_{\text{exact}}(2, 3) = p(1 - p)^2$ ($\{2, 3\}$ is acceptable, $\{1, 2\}$ and $\{1, 3\}$ are not). Algorithm 2 leads to the same values except $D(2, 3) = D_{\text{exact}}(2, 3) + p^3(1 - p)$. The approximation (2) is not accurate in this example, but it is quite close to (1) for small values of p. Since an exact computation becomes more and more complicated as the number of peers increases, we need simulations to validate Assumption 1 in more general situations (see Section 6.4).

Correctness of Formula (2) We assume from now on that D is defined by equation (2) instead of (1), so D is now an approximation of probability events. Theorem 1 verifies that Lemma 2 still holds with the approximate formula.

Theorem 1 $\forall p \in]0, 1], \forall i \in \mathbb{N}^*, \sum_{k=1}^{\infty} D(i, k) = 1$.

Proof of Theorem 1 $D(i, j)$ no longer has the interpretation of a probability of an event. Using formula (2) we prove first that $S_i(j) := \sum_{k=1}^{j} D(i, k) \leq 1$, and then that $S_i(j) \xrightarrow[j \to \infty]{} 1$.

First part is proved by recurrence: $S_1(1) = p \in [0, 1]$. Suppose the bound is verified for $i + j \leq K$. For i, j such that $i + j = K + 1$, $S_i(j) = S_i(j - 1) + D(i, j) = S_i(j - 1) + (1 - S_i(j - 1))x$, with $x = p(1 - S_j(i - 1)) \in [0, 1]$, entailing that $S_i(j) \in [0, 1]$.

Then we show the limit is 1. If not, there exists some $\epsilon > 0$, such that $\sum_{k=1}^{\infty} D(i, k) < 1 - \epsilon$. If we put this back in formula (2), then:

$$D(i, j) \geq p\epsilon \left(1 - \sum_{k=1}^{i-1} D(k, j)\right) \quad (3)$$

We know that $\sum_{k=1}^{\infty} D(i, k)$ has a finite limit, thus $D(i, j) \to 0$ when $j \to \infty$. From equation (3) it follows that $\sum_{k=1}^{i-1} D(k, j) \to 1$. A particular consequence is that for all j large enough: $\sum_{k=1}^{i-1} D(k, j) \geq \frac{1}{2}$, which is impossible since the $i - 1$ sequences $(D(k, j))_{1 \leq k < i; j = 1..\infty}$ converge towards 0.

6.2 Fluid limit

We present now some mathematical consequences of assumption 1. When the number of peers is large, the model scales and the normalized histogram of neighbors tends to a continuous distribution and yields an equation satisfied in this limit. Indeed the empirical distribution also converges, which means that every instance of an Erdös-Rényi graph is very likely to behave like the typical case of the above assumption, as shown by the simulations below.

We are able to prove some parts of this program but must leave the remainder as conjectures for further work. The results bring considerable insight.

From a practical point of view, the main result is that there exists a scaled version of D that converges towards a distribution as n increases.

- If p is fixed, we have a Dirac limit,
- if $d = p(n - 1)$ is fixed, the limit is a continuous distribution,
- the shape of $D(i, j)$ is present in almost any given n-peers system.

First, we define the scaled version of D: it consists in representing a peer i by a normalized ranking $0 \leq \alpha_n < 1$. The scaled version, denoted $\mathcal{D}$, is defined by $\mathcal{D}_{n,p}(\alpha, \beta) = nD_{n,p}(1 + \lfloor n\alpha \rfloor, 1 + \lfloor n\beta \rfloor)$. $\mathcal{D}_{n,p}$ is a simple function on $[0,1[^2$. Its range is the set of $(D_{n,p}(i,j))_{1 \leq i,j \leq n}$. The factor n in its definition allows to express $D(i,j)$ as an integral of $\mathcal{D}$: $D_{n,p}(i,j) = \int_{\frac{j-1}{n}}^{\frac{j}{n}} \mathcal{D}_{n,p}(\frac{i-1}{n}, x)dx$.

With this scaling notations, a first limit (Dirac limit) is given by Theorem 2.

Theorem 2

$$\forall p \in]0,1], \forall \alpha \in [0,1[, \mathcal{D}_{n,p}(\alpha, .) \xrightarrow[n\to\infty]{*} \delta_\alpha.$$

Theorem 2 expresses that when n increases and p is fixed, the scaled version of D has a Dirac limit: the normalized gap between a peer of a given normalized rank and its mate becomes arbitrarily small if n is big enough.

Conjecture 1 gives the other limit (fluid limit):

Conjecture 1 (Fluid limit) *Let $d \in \mathbb{R}^+$ be an average degree and $0 \leq \alpha < 1$ be a rank. We define $p_n := \frac{d}{n-1}$. There exists $\mu_{\alpha,d} \in \mathcal{P}([0,1])$ that is absolutely continuous with respect to Lebesgue measure such that:*

$$\mathcal{D}_{n,p_n}(\alpha, .) \xrightarrow[n\to\infty]{} \mu_{\alpha,d}.$$

Conjecture 1 states that if d is fixed, the normalized gap distribution weakly converges towards a continuous distribution.

Sketch of proof of Theorem 2: Existence of a weakly convergent subsequence is a standard tightness property on $[0,1]$: all the mass stays in a compact set and $\mathcal{D}_{n,p}(\alpha, .)$ has an increasing mass. For uniqueness, we admit that $D_{i+k,j+k}$ is decreasing wrt k. Then, for $\alpha < \beta$, we have

$$\begin{aligned} \mathcal{D}_{n,p}(\alpha,\beta) &= nD_{n,p}(1 + \lfloor n\alpha \rfloor, 1 + \lfloor n\beta \rfloor) \\ &\leq nD_{n,p}(1, 1 + \lfloor n\beta \rfloor - \lfloor n\alpha \rfloor) \\ &\leq np(1-p)^{\lfloor n\beta \rfloor - \lfloor n\alpha \rfloor - 1} \xrightarrow[n\to\infty]{} 0 \end{aligned}$$

This makes δ_α the unique possible weak limit.

Proof of Conjecture 1 for $\alpha = 0$: Let $\beta \in [0,1[$. $\mathcal{D}_{n,p_n}(0,\beta) = nD_{n,p_n}(1, 1 + \lfloor n\beta \rfloor) = np_n (1-p_n)^{\lfloor n\beta \rfloor - 1}$. This implies

$$\mathcal{D}_{n,p_n}(0,\beta) \sim d\left(1 - \frac{d\beta}{n\beta}\right)^{n\beta} \to de^{-\beta d}.$$

This in turn yields: $\mu_{0,d}(\mathrm{d}\beta) = de^{-\beta d}\mathrm{d}\beta$.

This theoretical result could be proved for $0 < \alpha < 1$ though at the expense of very long and technical developments. We do not anticipate any significant mathematical difficulty though it does remain to carry through the demonstrations. The results are not necessary to make the following observations, but they explain why we have considered some particular scaling.

6.3 Observations

The results in this section are obtained by solving Equation 2. We took $n = 5000$ to obtain the smoothest possible curves (because of the fluid limit, $n = 100$ would give quite similar results). d is set to 25. Figure 4 illustrates the different cases that may arise.

Figure 4(a) shows the case of a well-ranked peer. For $i = 1$ the right part is geometrically distributed. In average, the best 5% peers are peered with peers of lower rank. This changes quickly, and peers in the top 20% but not in the top 5% tend to get a significantly better mate.

The case of an average-ranked peer is illustrated in Figure 4(b). The distribution is symmetric; it simply shifts with the rank of the considered peer (for top 25% to top 80% peers). This second fact is a kind of finite horizon property and illustrates the property we called stratification. Notice that the distribution cannot be fit with a normal law, in any case.

In Figure 4(c), the distribution shift continues for the bottom 20% of peers, but as there is no worse peer to mate with, the distribution is cut. This means that there is a probability for not being matched (the area filled in blue). The lowest matching frequency is for the worst peer, which is matched exactly in half of the cases.

6.4 b_0-matching independent model

The 1-matching case gives a flavor of the stratification phenomenon. Formally there are no new issues in progressing to a b_0-matching model except for the weight of notation. As in the case of 1-matching, we state an independence assumption which is not formally true but provides a fairly good approximation compared to simulations.

Notation

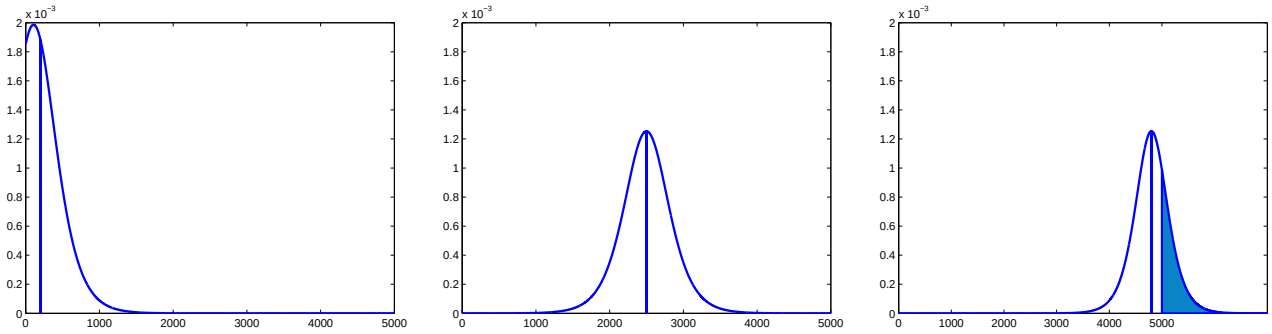
The situation becomes more complicated, because the first choice of one peer may correspond to the last choice of its mate. Consequently we have to study a quantity $D_c^{c'}(i,j)$ which is not of direct interest. This is the probability that choice number c of peer i is j and that for j, i is choice number c'. As in the 1-matching case, $D_c^{c'}(i,j)$ does not depend on larger indexes for i, j, c and c'. Nor does it depend on n. Intuitively this corresponds to the fact that the first choice is made before making the second, and that the best peers have priority for choosing their mates. The quantity of interest is the probability that choice number c of i is j: $D_c(i,j) = \sum_{c'=1}^{b_0} D_c^{c'}(i,j)$.

Independent b_0-matching algorithm

Hereinafter we shall adopt the following assumption:

Assumption 2 *Let $i, j \leq n$ and $c \geq 1$ and $c' \geq 1$. These two events are independent:*

- *peer i has chosen $c-1$ peers better than j and choice c is not a peer better than j,*



(a) $D(200,.)$ (b) $D(2500,.)$ (c) $D(4800,.)$

Figure 4. Distribution $D(i,.)$ for three values of i.

- *peer j has chosen $c'-1$ peers better than i and choice c' is not a peer better than i.*

Under Assumption 2, we evaluate $D_c^{c'}(i,j)$ by multiplying the probabilities of the assumed independent events: $\{i,j\}$ is an edge of the acceptance graph (probability p); choice c of i is not a peer better than j, but previous choices are; the reciprocal condition on j.

The probability that choice c of i is not a peer better than j, whereas previous choices are, is simply $\sum_{k=1}^{j-1} D_{c-1}(i,k) - \sum_{k=1}^{j-1} D_c(i,k)$: the probability that choice $c-1$ of i is a peer better than j minus the probability that choice c of i is a peer better than j (this formula is mathematically exact because one of the two events is included in the other). This proves, under assumption 2, that:

$$D_c^{c'}(i,j) = p\left(\sum_{k=1}^{i-1} D_{c'-1}(j,k) - D_{c'}(j,k)\right) \times \times \left(\sum_{k=1}^{j-1} D_{c-1}(i,k) - D_c(i,k)\right). \quad (4)$$

Algorithm 3 show how to compute this formula by dynamic programming.

Remark that whereas $D_c(i,j)$, the c-th choice distribution of i is no longer symmetric for $b_0 > 1$, $D_c^{c'}(i,j)$ has more symmetry (see Algorithm 3). Matlab scripts for this alhorithm can be found at [9]. This version is not optimized (but sufficiently fast for the needs of this paper).

Validation of independent b_0-matching As mentioned above, Assumptions 1 and 2 are approximations, but they should work very well except for very small numbers of peers with p very large. Figure 5 illustrates this point. We simulated a 2-matching by drawing a million realizations of the Erdös-Rényi graph with $n = 5000$ and $p = 1\%$ (simulations requiring several weeks) and compared obtained distributions $D_1(3000,.)$ and $D_2(3000,.)$ with those given by our simplified formula. The results confirmed the accuracy of the formula, as illustrated by Figure 5.

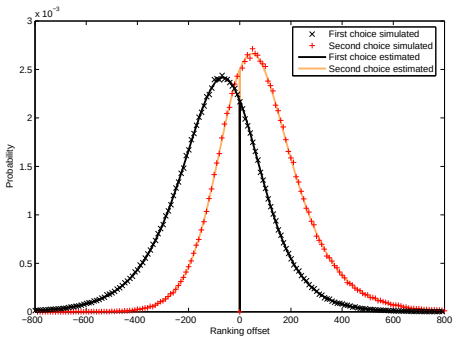


Figure 5. Exact(simulated) and estimated (using (4)) values of $D_1(3000,.)$ and $D_2(3000,.)$.

7 Application to BitTorrent

Results of previous Sections allow us to closely estimate for each peer, the ranks of peers it is likely to collaborate with. All our results tend to give a theoretical proof of the stratification phenomenon in systems that use a global ranking function. In this Section, we will see how this stratification can give insight into the effect of the Tit-for-Tat policy used in BitTorrent.

We suppose that we are in the *steady-state* phase (after flashcrowd). In the *flashcrowd* phase, a unique seed is uploading a new file, and the upload capacities of the best peers are useless: all peers have downloaded the same blocks. But during the post-flashcrowd phase, all blocks have roughly the same dispersion, because of the download-rarest-first policy of BitTorrent. So we can assume that content availability will not affect the acceptance graph and focus on bandwidth only.

The TFT policy consists in uploading to the peers from which one gets the best download rates. The selection process is renewed periodically. Furthermore, a generous upload connection allows to probe new peers for an eventual TFT exchange. This protocol acts like the peer initia-

Algorithm 3: Independent b_0-matching probability

Data: Erdös-Rényi parameters n, p
matching quota b_0
Result: $D_c^{c'}(i,j)$ the probability that the c-th choice of peer i is j and that the c'-th choice of j is i,
$D_c(i,j)$, probability that the c-th choice of peer i is j

$D_c \leftarrow zeros(b_0, n, n)$
$D_c^{c'} \leftarrow zeros(b_0, b_0, n, n)$
$D_0^c \leftarrow ones(1, b_0, n, n)$
$D_c^0 \leftarrow ones(b_0, 1, n, n)$
for $i = 1$ **to** n **do**
  for $j = i + 1$ **to** n **do**
    for $(ci, cj) \in [|1, b_0|] \times [|1, b_0|]$ **do**
      Compute $D_c^{c'}(i,j)$ using (4)
    for $c = 1$ **to** b_0 **do**
      $D_c(i,j) \leftarrow \sum_{c'=1}^{b_0} D_c^{c'}(i,j)$
    for $c' = 1$ **to** b_0 **do**
      $D_{c'}(j,i) \leftarrow \sum_{c=1}^{b_0} D_c^{c'}(i,j)$

tive described in Section 2. We thus claim that our results apply to the TFT exchanges in BitTorrent. In particular, we have a proof of the stratification effects (peers tend to exchange with peers with similar bandwidths) empirically observed by [1, 6].

However, the ranking of a peer just gives an intuition about the Quality of Service (QoS) it is presumed to experience. In order to obtain relevant results, it is necessary to bind ranking and performance. In the case of a file sharing system like BitTorrent, the average expected download rate is a very convenient performance metric, especially since it is easy to compute within our model: it is enough to know the upload bandwidth for each peer i.

To compute network performances, we have taken as reference the measurements made by Saroiu *et al.* [10]. Using bandwidth estimation in the Gnutella network, they have estimated the upstream for a large community of P2P users. The cumulative distribution they obtained is shown Figure 6. One can observe a wide distribution of bandwidths (just like in Orwell's *Animal Farm*, "all peers are equal but some peers are more equal than others").

Applying our fluid model to the distribution observed by Saroiu *et al.*, we get the results shown in Figure 7. We chose the following parameters:

- b_0-matching with $b_0 = 3$, corresponding to a BitTorrent network with all clients having the default number of slots of 4 (the fourth is the "generous slot" used to probe connections, not for tit-for-tat policy).
- expected number of acceptable peers (peers who are known and interesting) $d = 20$ (realistic value)

Notice that the number n of peers does not have to be given because our model does not depend on the network size: with a partial network knowledge, observed offsets scale with the number of peers (see Section 6.1).

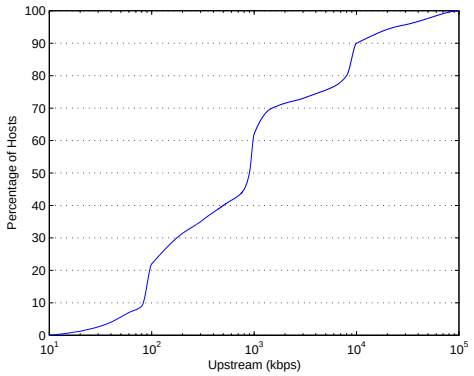


Figure 6. Upload capacity CDF (from [10]).

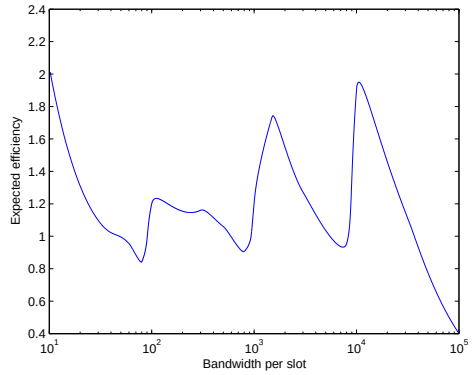


Figure 7. Expected D/U ratio as a function of the upload.

In order to present clear results, we chose to represent expected download/upload ratio, which correspond to BitTorrent share ratio. When this ratio is less than 1, a peer gives on average more that it receives.

Some worthwhile observations are as follows:

- Best peers suffer from low sharing ratios: as they are the best, they can only collaborate with lower peers, so the exchange is suboptimal for them. The only way for best peers to counter this effect is by adding extra connections until the upload bandwidth per slot is close to the one of lower peers. This somehow explains why BitTorrent proposes *by default* a greater number of connections (up to TCP limitations) for peers with high bandwidths, thus avoiding too much spoil.
- There are density peaks in the bandwidth distribution. This peaks corresponds to typical Internet connections, such as DSL or cable. Peers in the density peeks have a ratio close to 1. This is due to the great probability they have to collaborate with peers that have exactly the same characteristics as them.

- Efficiency peeks appear for peers that have an upload just above a density peek. For these peers, lower peers have almost the same upload bandwidth as them, whereas upper peers are likely to offer greater bandwidth.
- Surprisingly, the lowest peers have a high efficiency, although there is some probability for them not to be matched (see Fig. 4(c)). This is related to the high bandwidth (compared to their) they sometimes obtain, which overcompensate the probability to be rejected.

As a consequence of this non-uniform efficiency distribution, it is tempting for an average peer to tweak its number of connections in order to increase the efficiency of its connections. For instance, suppressing one connection can improve the probability of collaborating with higher peers. However, this leads to a Nash equilibrium where all peers have just one TFT slot. This is unacceptable in terms of connectivity, but rational peers trying to maximize their benefit cannot be avoided. This is an explanation for the 4 slots (3 TFT and one generous slot) settings: obedient average peers that uses the default settings must have at least 4 in order to ensure connectivity in the TFT collaboration graph. On the other hand, the more slots they have, the farther they are from the Nash equilibrium that rational peers will try to follow. Hence 4 seems to be the best trade-off.

8 Conclusion

In this paper, we identified the stable matching theory as a natural candidate to model peer-to-peer networks where peers choose their collaborators. Furthermore, we applied elements of this theory to a specific case: b-matching with global rankings. While there has been a lot of work in analyzing incentives to collaborate in some specific application from an economical point of view, this is the first attempt to analyze the behavior of a class of applications using graph theory.

The main conclusion of this study is that matching theory gives insights on the behavior of a P2P systems class, namely the global ranking class. For both the case of complete acceptance graphs and the case of random acceptance graphs, we studied clustering and stratification issues. In most cases, clustering may be prevented using b-matching with enough connections and some standard deviation. But stratification is an intrinsic property of such networks. It seems impossible to overcome it as long as each peer follows the *try-to-collaborate-with-the-best* rule. Interestingly, for random overlay graphs, the crucial parameter is d, the average number of acceptable peers, which makes stratification a flawlessly scalable phenomenon.

As a first application, our results provide some new insights on BitTorrent parameters. They show that best peers have to set up a large number of connections in order to avoid a bad download/upload ratio. The *by default* number of collaborations (4) is justified. It allows, to a certain extend, to maintain connectivity in the TFT exchanges and to protect peers using default settings (obedient peers) from peers with optimized settings (rational peers).

When considering the stable properties which emerge, it also becomes clear that different classes of utility functions lead to very different properties. This can be exploited according to the needs of the targeted application. For example, in a peer-to-peer streaming protocol, the most important feature is a small playout delay but a strong stratification, needed to give peers incentive to collaborate, produces a collaboration graph with a large diameter (large playout delay). In many cases, combining different utility functions will be necessary. Such a combination can, for instance, be achieved by introducing a second type of collaborations depending on a different global ranking or depending on a symmetric ranking such as latency.

Acknowledgment: *The authors wish to thank James Roberts, Nidhi Hegde and Dmitri Lebedev for their helpful comments*

References

[1] A. R. Bharambe, C. Herley, and B. N. Padmanabhan. Analyzing and improving a bittorrent networkś performance mechanisms. In *Proceedings of IEEE Infocom*, 2006.

[2] K. Cechlárová and T. Fleiner. On a generalization of the stable roommates problem. *ACM Trans. Algorithms*, 1(1):143–156, 2005.

[3] B. Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[4] http://www.edonkey2000.com/index.html.

[5] D. Lebedev, F. Mathieu, L. Viennot, A.-T. Gai, J. Reynier, and F. de Montgolfier. On using matching theory to understand P2P network design. In *INOC*, 2007.

[6] A. Legout, N. Liogkas, E. Kohler, and L. Zhang. Clustering and sharing incentives in bittorrent systems, 2006.

[7] F. L. Piccolo, G. Neglia, and G. Bianchi. The effect of heterogeneous link capacities in bittorrent-like file sharing systems. In *HOT-P2P '04: Proceedings of the 2004 International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P'04)*, pages 40–47, Washington, DC, USA, 2004. IEEE Computer Society.

[8] D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks, 2004.

[9] J. Reynier. Erdös-rényi pairing in matlab. http://www.di.ens.fr/ jreynier/Recherche/MatlabR2.zip .

[10] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking*, 2002.

[11] J. J. M. Tan. A necessary and sufficient condition for the existence of a complete stable matching. *J. Algorithms*, 12(1):154–178, 1991.